XAI for Semantic Dependency

How to understand the impact of higher-level concepts on AI results

Holger Ziekow
Faculty of Business Information Systems
Furtwangen University
Germany
e-mail: holger.ziekow@hs-furtwangen.de

Peter Schanbacher
Faculty of Business Information Systems
Furtwangen University
Germany
e-mail: peter.schanbacher@hs-furtwangen.de

***Abstract*—XAI methods such as partial dependency plots or individual conditional expectation plots help understanding the impact of feature values on the output of an AI model. However, these techniques can only analyze the concepts manifested in a single feature. This makes it hard to investigate the impact of higher-level concepts, spanning across multiple features (E.g. a model prediction may depend on the morbidity of a patient, while morbidity is only indirectly reflected through features about symptoms). In this paper we present and test a concept for getting insight into model dependency on aspects on a higher semantic level. This enables an understanding how a model output changes in dependence on meaningful higher-level concepts and aids data scientists in analyzing machine learning models.**

Keywords-Interpretability, Understandability; Explainability; explainable AI; XAI; human-centered AI; black-box models

I. Introduction

Due to increasing computational power, improving algorithms and access to big-data, Artificial Intelligence (AI) models gained popularity in recent years. Applications range from healthcare (Lee et al., [15]; Chen et al., [6]), credit risk (Szepannek and Lübke, [23]), autonomous driving (Grigorescu et al., [14]; Feng et al., [9]), image classifications (Sahba et al., [20]), audio processing (Panwar et al, [19]), among others.

The large number of parameters and complex interactions makes most AI models (in particular deep neural networks) hard to understand and difficult to interpret the results. For many applications it is required not only to have a model with high accuracy but also explain the outcomes. Regulators (European Commission, [8]) require the understandability of these models, in particular to increase their trust (Lui and Lamb, [17]) and assess potential biases (Challen et al., [5]).

What "explainability" means is not well defined and might be misleading (Rudin, [27]). It further depends on the context of the application. For MRI scan, the explanation might be a heat map of relevant areas for the model. For sentiment analysis of user feedback, the explanation might be relevant words of the text. Surrogate models such as decision trees may give an insight into more complex models.

In general, explainability methods can be distinguished into either global explainability on the model level such as variable importance (Breiman, [4]), partial dependency plots (PDP, Friedman, [10]), or accumulated local effects (ALE, Apley and Zhu, [1]), or local explainability on the level of individual predictions such as Shapley values (SHAP, Shapley, [21] or Strumbelj and Kononenko, [22]), or local interpretable model explanations (LIME, Ribeiro et al., [24]).

We lean on the notion of partial dependency plots (PDP). However, unlike PDPs, we capture the dependency on a higher-level concept, and not a single feature. (E.g. a concept that manifests in many features or the combination of many feature values). The analysis shows the model output if a certain concept is more or less present. E.g. one may analyze if a medical model leans more or less towards a certain recommendation, dependent on the morbidity of a patient. Yet, the morbidity may not be an explicit input of the model but indirectly reflected in a set of features about certain symptoms. Another example is an image classifier. Existing methods analyze the impact of pixels or regions in specific figures (see e.g. Bulat & Tzimiropoulos, [3]). However, reasoning about the semantics of these regions is up to the analyst and must be done instance by instance. With our method, one gains an understanding how presence of a certain concept impacts the model output. To the best of our knowledge, this constitutes a new approach. In this context, we refer to the approach as semantic dependency analysis (not to be confused with semantic dependency in NLP). As an illustrative example, we analyze how the presence of vegetation impacts the classification of an image as showing a city or rural area.

Our main contributions are the following

- We present a new general concept which we call semantic dependency analysis (SDA).
- We provide formalisms to define two fundamental ways of implementing SDA.
- We describe a specific implementation along a sample case.
- We present experimental results that demonstrate the working and utility of the approach.

The remainder of the paper is structured as follows: The introduction is followed by section 2 presenting the current state of literature and how our approach fits into the related work. Section 3 defines the concept of Sematic dependency analysis (SDA) and presents a possible implementation for generators as well as prediction models. Section 4 shows how

SDA can be used for an illustrative image classification example. Section 5 summarizes and concludes.

II. RELATED WORK

White box models, also known as transparent or interpretable models, offer humans a clear understanding of the underlying decision-making process. White box models are algorithms such as linear regression, decision trees, or logistic regression. On the other hand, there are black box models such as deep neural networks. They have a vast number of parameters and can therefore account for complex interactions. While these models often exhibit remarkable performance, their decision-making processes is difficult to understand. This lack of interpretability raises concerns regarding trust, fairness, accountability, and potential biases within the model (see Riberio et al., [24]). Explainable Artificial Intelligence (XAI) is needed to establish trust of the user and the AI model (Arrieta et al., [2]). Users want to have information why the model proposed a certain decision (Wang et al., [28] or Gandi and Mishra, [12]). Further XAI is needed to detect and mitigate biases to promote fairness (Ridley, [25]). Recent regulation requires the “right to explanation” for individuals affected by AI-driven decisions (Gallese, [11]). Another prominent application area of XAI is the medical domain, due to the often sensitive nature of AI decisions. (see [29] for a survey). PDPs (see Friedman, [10]) have long been used to understand the impact of a certain feature. PDPs have computational advantages and are easier to understand for a layman compared to most alternative XAI methods (Dwivedi et al., [7]). However, PDPs do not properly take feature interactions into account (Linardatos et al., [16]). To account for the interaction effects, individual conditional expectation plots (ICE, see Goldstein et al., [13]) were developed. An alternative approach are Accumulated Local Effect plots (ALE, Apley and Zhu, [1]). While PDPs are based on the marginal distribution, ALE plots are based on the conditional distribution. All those PDPs related methods, show the impact of a certain feature given in the dataset. Higher-level concepts which are often of interest but not included in the data, therefore cannot be analyzed. Consider for example patient data such as age, sick days, therapy, income. The higher-level concept of interest “morbidity” is however not the data. To analyze the impact of such a higher-level concept we introduce the semantic dependency analysis.

III. SEMANTIC DEPENDENCY ANALYSIS

In this section, we introduce the concept of semantic dependency analysis (SDA). We lean on the notion of partial dependency plots that are defined as follows (see Molnar, [18]):

$$\hat{f}_S(x_S) = E_{X_C}\left[\hat{f}(x_S, X_C)\right] = \int \hat{f}\,(x_S, X_C)dP(X_C).$$

Here, x_S is the feature value of the analyzed feature S, X_C are the other features in the model, and $\hat{f}(x_S, X_C)$ the AI model applied on the complete feature vector (containing x_S and X_C). Intuitively, the partial dependence function represents the average prediction if all data points have the given feature value x_S.

In SDA, we do not analyze a single feature S but a higher-level concept H where $x_H \in H$ are values reflecting the presence (or degree of presence) of that concept (i.e. for elements in H we expect an order relation with respect to the presence of the semantic concept H). We define the analysis for a given higher level concept H (SD_H) as

$$SD_H(x_H) = E_X\left[\hat{f}\left(g(x_H, X)\right)\right].$$

Here $g(x_H, X)$ is a random variable that returns feature vectors for the model $\hat{f}$ in accordance to x_H , and in compliance with X. That is, the resulting values stem from the distribution of X and also have concept x_H. We subsequently discuss two concepts of the implementation of g.

A. Implementation with generators

One way to implement g is to use synthetic data generators. Values of x_H in H and the distribution of X drive the generation of data points in accordance to x_H . For instance, x_H may be mapped to a prompt in a text to image model. The distribution of X may be reflected by further elements of the prompt (i.e. a prompt describing X). Note that there are further options to account for the distribution of X. This includes the use of image-to-image models, taking samples from X as input, or training the generator on X.

Alternative implementations may use rule-based data generators (in particular for tabula data) or 3D engines for image generation. The feasibility of different data generation approaches depends heavily on the use case. In our sample case, we use a diffusion model for illustration (see section 4).

B. Implementation with prediction models

Another way of implementing g is to use a prediction model $d(x, x_H)$ that can detect the presence of x_H in a data point x in X. The advantage is that data points can be sampled from real data with the distribution of X. Assuming that d returns a score for the presence of x_H in x, we can implement $g(x_H, X)$ by drawing from $\{x \in X \mid d(x, x_H) \geq t\}$, where t is a threshold denoting the minimum probability that x_H is present. Note, that using d also allows for an alternative definition of SD_H as continuous function, dependent on the ε-environment around the presence score s , denoting the presence of X_H in a data point x:

$$SD_H(x_H, s) = E_X\left[\hat{f}(\{x \mid d(x, x_H) \in [s - \varepsilon, s + \varepsilon]\})\right].$$

IV. EXPERIMENTS

In this section, we describe experiments that demonstrate along an example the viability of the approach and illustrate a possible instantiation of the concept.

A. Experimental Setup

We use a sample application as proof of concept and test for the SDA approach. As sample task, we use a binary classification task for images. That is, we aim to classify images in either class A (showing a city) or class B (showing a rural landscape).

For our test application we useed synthetic data, generated with stable diffusion version 2 (see Rombach et al., [26]). We used default parameters, except for the sampling steps of 100. The prompts for generating the different classes are:

Class A:

Positive prompt: *Photograph a city, high quality photography, Canon EOS R3*

Negative prompt: *digital art, drawing*

Class B:

Positive prompt: *Photograph of a rural landscape, high quality photography, Canon EOS R3*

Negative prompt: *digital art, drawing*

For the training data, we generated images of 512×512 pixels. Figure 8 and 9 show examples from the training set of both classes. For the experiments, we use an arbitrary classification network generated by ChatGPT 4.0. The network architecture is shown in Figure 1. In each test run, we trained the model using 5 epochs and batch size 32. We then analyzed the resulting networks with SDA.

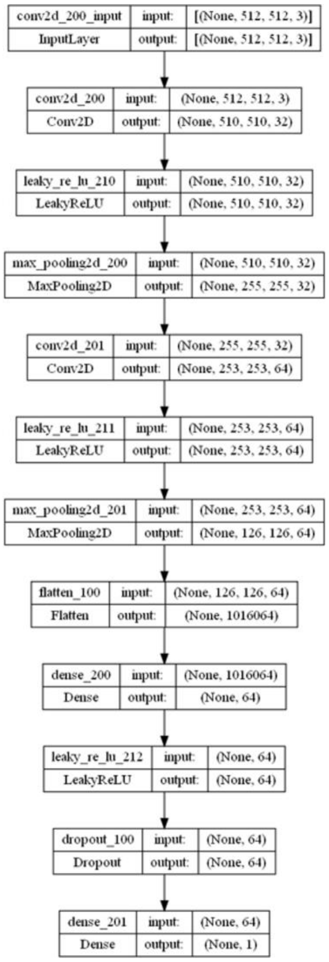

Figure 1 Network architecture for experiments

For the SDA, we implement $g(x_H, X)$ by adding to the prompt of class A. That is, we limit the analysis to the recognition of class A. By keeping the prompt for class A, we realize the compliance to X in the data generation. Note that this is only an approximate solution, as the control over the diffusion model's output is limited. By adding to the prompt of A, we implement the presence of x_H. By using in total 4 different prompt additions, we implement an order over the total of 4 elements x_H in H. Specifically, we used the following prompts.

Data set "cityNoTrees":

Same prompt as for class A but with "trees" in negative prompt. See Figure 10. Images in this data set contain some trees but less than the data sets "City", "cityTrees", "TreesCity".

Positive prompt: *Photograph of a rural landscape, high quality photography, Canon EOS R3*

Negative prompt: *digital art, drawing, trees*

Data set "City":

Same prompts as for as class A. Images in this data set contain more trees than the data sets "cityNoTrees", but less than in "cityTrees" and "TreesCity".

Data set "cityTrees":

Same as class A but with "trees" added after "city" to the positive prompt (giving more importance to "city" than to "trees"). See Figure **11**. Images in this data set contains more trees than the data sets "cityNoTrees", and "City", but less than in "TreesCity".

Positive prompt: *Photograph a city, trees, high quality photography, Canon EOS R3*

Negative prompt: *digital art, drawing*

Data set "TreesCity":

Same as class A but with "trees" added before "city" to the positive prompt (giving more importance to "trees" than to "city"). See Figure **12**. Images in this data set contains more trees than all the other data sets, but are still generated to show locations within cities.

Positive prompt: *Photograph trees, city, high quality photography, Canon EOS R3*

Negative prompt: *digital art, drawing*

The prompts for the different data sets are given in the order of presence of the concept "trees". Although the exercised control over the diffusion networks is limited, manual inspection of the resulting images verifies the intended outcome. (The test set for "city" is not shown, as the prompt was the same as for the training set and the results are of similar appearance).

We trained 6 different networks with the given architecture on the training set containing classes A and B. (Choosing 6 networks is merely due to space limitations in the paper). We used 800 images for each class. For the SDA we used 800 images for each of the 4 prompts "cityNoTrees", "City", "cityTrees" and "TreesCity". With the SDA we aim to investigate if the presence of the concept "trees" makes the model less confident about class A; That is, if increased presence of trees leads to a lower probability for detecting the class A (city).

B. *Results*

Figures 2-7 show results of the tests for six different networks. The results show box plots of the probabilities assigned by different models to the images for the data sets “cityNoTrees”, “City”, “cityTrees” and “TreesCity”. The probability shown is the determined probability of *not* showing a city. The mean values in the box plots are the values intended for the SDA, according to our definition in section III. However, the additional information from the box plots gives additional insights about the distributionn.

If the presence of trees causes models to deem the label “city” less likely, we expect to see increasing means from left to right in the plots. That is because the “cityNoTrees”, “City”, “cityTrees” and “TreesCity” are ordered according to that presence of the higher-level concept “Tree”. We observe that this is the case for all analyzed networks. (Note that, although “City” used the same prompt as the training set, the models have an even higher confidence for “cityNoTrees“ than for “City”). Analysts learn from the SDA that the trained models are impacted by the higher-level concept of trees as well as the nature of that impact. The behavior of the networks and the SDA results are plausible. Hence, the experiments verify the viability and utility of our approach.

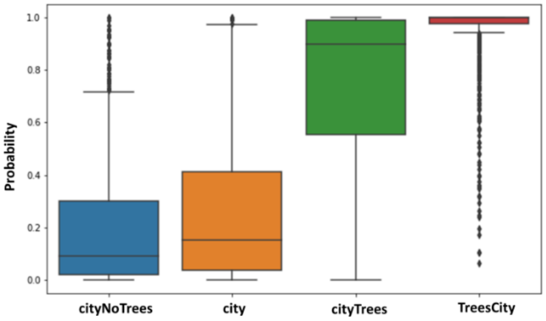

Figure 2 SDA for network 1

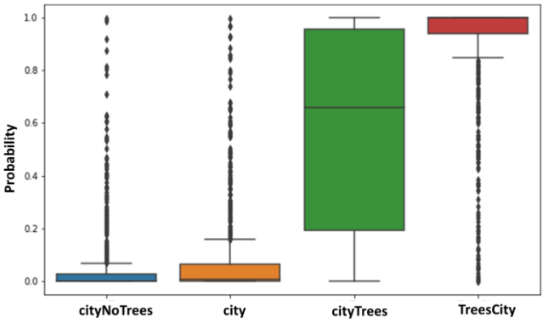

Figure 3 SDA for network 2

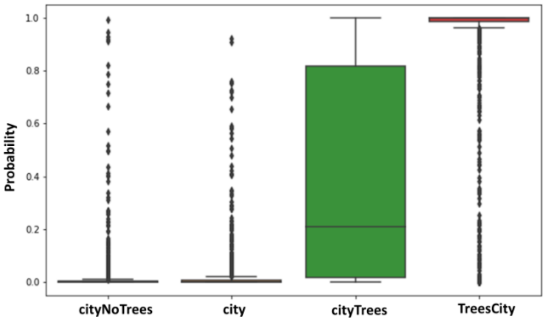

Figure 4 SDA for network 3

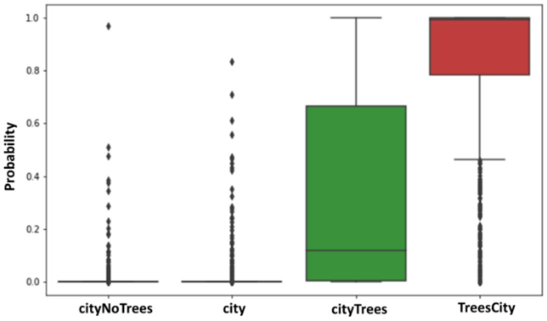

Figure 5 SDA for network 4

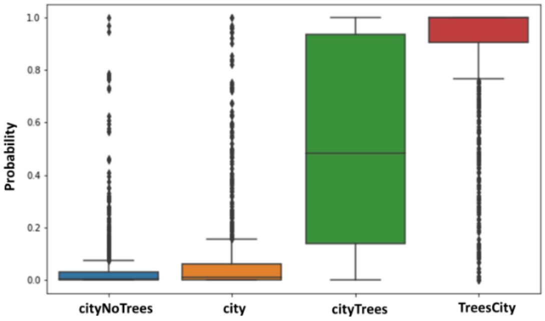

Figure 6 SDA for network 5

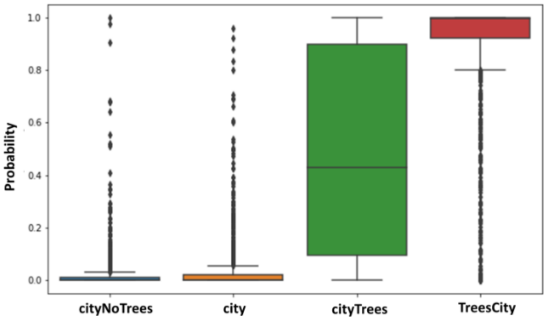

Figure 7 SDA for network 6

V. CONCLUSIONS

Our article introduces the concept of semantic dependency analysis (SDA), which goes beyond traditional partial dependency plots (PDPs) by capturing dependencies on higher-level concepts rather than individual features. The analysis showcases how the output of a model changes based on the presence or absence of a specific concept.

As the illustrative example, we analyzed how the presence of vegetation affects the classification of an image as a city or rural area. The results of the analysis demonstrate that the trained models are influenced by the higher-level concept of trees and provide insights into the nature of this impact. The observed behavior of the networks aligns with expectations and supports the viability and utility of the approach employed in the experiments. Analysis can use such insight to reason about the working of their models.

Future work will address further options for implementing $g(x_H, X)$ and the challenge that implementations may only approximate the intended behavior. In particular with the generative approach, reflecting x_H to the desired degree is a challenge in implementing g. However, our illustrative example shows the viability.

By moving beyond individual features and focusing on broader concepts, SDA provides valuable insights into how a higher-level concept influences predictions or classifications. The formalisms and implementation described in the text provide a foundation for conducting SDA and analyzing various domains, ranging from medical models to image classifiers. Overall, SDA has the potential to enhance interpretability and decision-making in AI systems, contributing to advancements in explainable AI and fostering trust in AI-driven solutions.

ACKNOWLEDGEMENTS

We like to thank Valentin Göttisheim for feedback and support in the creation of this paper.

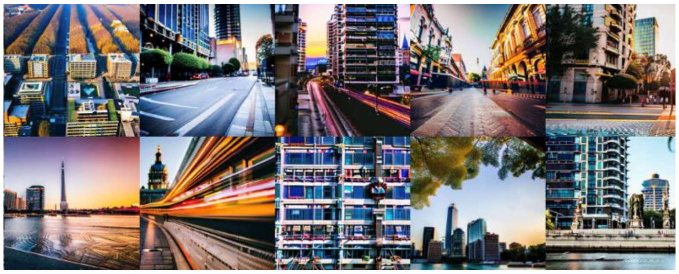

Figure 8. Samples from training set for label "City"

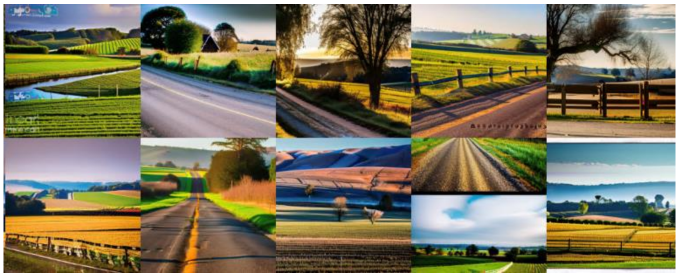

Figure 9 Sample from training data set for label “Landscape”

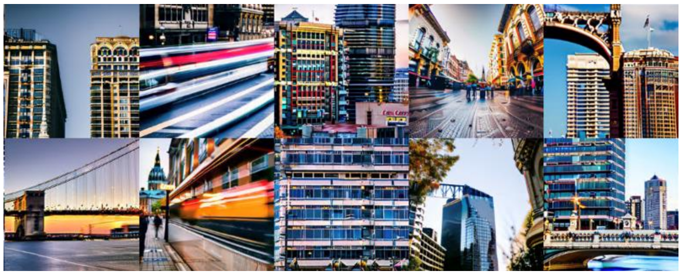

Figure 10 Sample from test set with “city” in positive and “trees” in negative prompt (cityNoTrees)

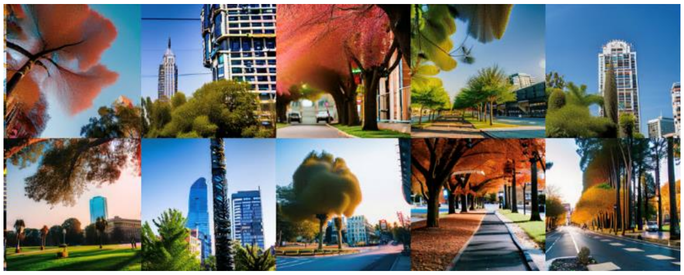

Figure 11 Sample from test set with first “city” and then “trees” in positive prompt (cityTrees)

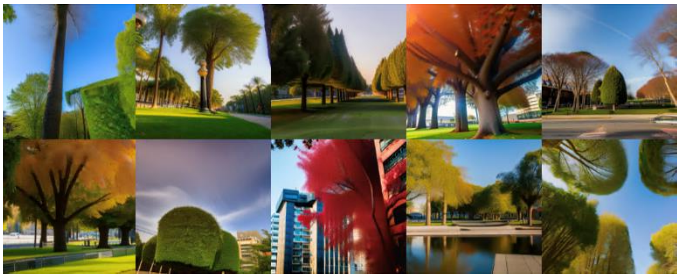

Figure 12 Sample from test set with first “trees” and then “city” in positive prompt (TreesCity)

REFERENCES

[1] Apley, D.W.; Zhu, J. (2020): Visualizing the effects of predictor variables in black box supervised learning models. J. R. Stat. Soc. Ser. B

[2] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A. & Herrera, F. (2020): Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion, 58, 82-115.

[3] Bulat, A., & Tzimiropoulos, G. (2016): Human pose estimation via convolutional part heatmap regression. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14 (pp. 717-732). Springer International Publishing.

[4] Breiman, L. (2001): Random Forests. Machine Learning, 45(1), 5-32.

[5] Challen, R., Denny, J., Pitt, M., Gompels, L., Edwards, T., & Tsaneva-Atanasova, K. (2019).

Artificial intelligence, bias and clinical safety. BMJ Quality & Safety, 28(3), 231-237.

[6] Chen, R. , Yang, L., Goodison, S. and Sun, Y. (2020): “Deep-learning approach to identifying cancer subtypes using high-dimensional genomic data,” Bioinformatics, vol. 36, no. 5, pp. 1476–1483

[7] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., ... & Ranjan, R. (2023). Explainable AI (XAI): Core ideas, techniques, and solutions. ACM Computing Surveys, 55(9), 1-33

[8] European Commission (2021): proposal for a Regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206 (access on 12.6.2023)

[9] Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., & Dietmayer, K. (2020). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems, 22(3), 1341-1360.

[10] Friedman, J. (2001): Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29, 1189-1232.

[11] Gallese, C. (2023). The AI Act proposal: a new right to technical interpretability?. arXiv preprint arXiv:2303.17558.

[12] Gandhi, N., & Mishra, S. (2022). Explainable AI for healthcare: A study for interpreting diabetes prediction. In Machine Learning and Big Data Analytics (Proceedings of International Conference on Machine Learning and Big Data Analytics (ICMLBDA) 2021) (pp. 95-105). Springer International Publishing.

[13] Goldstein, A.; Kapelner, A.; Bleich, J.; Pitkin, E. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. J. Comput. Graph. Stat. 2015, 24, 44–65.

[14] Grigorescu, S., Trasnea, B., Cocias, T., & Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. Journal of Field Robotics, 37(3), 362-386.

[15] Lee, SM, Seo, JB, Yun, J, Cho, Y-H, Vogel-Claussen, J, Schiebler, ML, Gefter, WB, van Beek, E, Goo, JM, Lee, KS, Hatabu, H, Gee, J & Kim, N (2019): Deep Learning Applications in Chest Radiography and Computed Tomography: Current state of the Art', Journal of Thoracic Imaging, vol. 34, no. 2.

[16] Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. Entropy, 23(1), 18.

[17] Lui, A., & Lamb, G. W. (2018). Artificial intelligence and augmented intelligence collaboration: regaining trust and confidence in the financial sector. Information & Communications Technology Law, 27(3), 267-283.

[18] Molnar, Christoph (2020): Interpretable machine learning. Lulu.com.

[19] Panwar, S., Das, A., Roopaei, M., & Rad, P. (2017, June). A deep learning approach for mapping music genres. In 2017 12th System of Systems Engineering Conference (SoSE) (pp. 1-5). IEEE.

[20] Sahba, A., Das, A., Rad, P., & Jamshidi, M. (2018, June). Image graph production by dense captioning. In 2018 World Automation Congress (WAC) (pp. 1-5). IEEE.

[21] Shapley, L. S. (1951): Notes on the n-Person Game. Santa Monica, RAND Corporation.

[22] Strumbelj, E. and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. Knowledge and Information Systems, 41(3), 647-665.

[23] Szepannek, G., Lübke, K. (2023). How much do we see? On the explainability of partial dependence plots for credit risk scoring. Argumenta Oeconomica, 1(50), 137-150.

[24] Ribeiro, M., Singh, S. and Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135-1144.

[25] Ridley, M. (2019): Explainable AI (XAI): Confronting Bias, Discrimination, and Fairness in Machine Learning. In Access Conference Proceedings.

[26] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022): High-Resolution Image Synthesis With Latent Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684-10695

[27] Rudin, C. (2019): Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” Nature Machine Intelligence, vol. 1, no. 5, pp. 206–215

[28] Wang, Y. C., Chen, T. C. T., & Chiu, M. C. (2023). An improved explainable artificial intelligence tool in healthcare for hospital recommendation. Healthcare Analytics, 3, 100147.

[29] Sheu, R. K., & Pardeshi, M. S. (2022). A Survey on Medical Explainable AI (XAI): Recent Progress, Explainability Approach, Human Interaction and Scoring System. Sensors, 22(20), 8068.

ISBN: 978-1-68558-111-4